Title: A Contour Classifying Kalman Filter Based On Evidence Theory

Authors: Sebastian Ohl

Markus Maurer

Institute: Braunschweig: Institute of Control Engineering

Published: 16 February 2012

url: http://www.digibib.tu-braunschweig.de/?docid=00042666

Original Digital Object Identifier: 10.1109/ITSC.2011.6082816

A Contour Classifying Kalman Filter Based On Evidence Theory

Sebastian Ohl and Markus Maurer

Abstract— **In the project Stadtpilot, introduced in [1], the object based environment perception system developed by the urban challenge team CarOLO at Technische Universität Braunschweig, as presented in [2], has been enhanced. The context of this new project is more challenging as now because it includes public traffic on large inner-city loops.**

Other vehicles are described by the project's sensor data fusion by an open polyline (contour) with many points. Some of these points lie on straight lines or they represent noise of the contour which do not contribute to the object's description. These extra points complicate an effective tracking and deform the contour of the object hypothesis. Because of the numerous traffic and due to the change in the environment's type, surrounded vehicles very often create a change of view. This results in no or less measurement updates of some points in the contour and can result in its deformation.

In an effort to overcome this problem, the contour estimating Kalman filter, presented in [3], has been enhanced by improved point update algorithms as well as a contour classifier based upon evidence theory. These enhancements allow the decrease of the used points. Changes of view, due to passing traffic, are better identified because the classifier identifies the most likely shape explicitly.

I. Introduction

Initially, the CarOLO project by Technische Universität Braunschweig adopted test scenarios containing locked test tracks with simulated traffic driven by stunt drivers. In the follow-up project, known as Stadtpilot, the test tracks have been extended to include public traffic on inner-city roads. In summer 2010 the test vehicle has successfully passed the first test runs with autonomous longitudinal and lateral track on Braunschweig's city ring.

In this new test track, the experimental vehicle's velocity is raised from approximately 30 km/h to up to 50-60 km/h and the number of other road users is significantly increased. Contrary to the contests of the DARPA urban challenge, motorbikes and trucks are also part of the test scenario.

To sense the vehicle's environment, the experimental vehicle has been equipped with many sensors for environment perception. These sensors are fused by an object based sensor data fusion algorithm to form a consistent view of the vehicle's surroundings. They are passed on to the application by an uniformed interface. By tracking already known object hypotheses with an observer, unmeasured states like acceleration can be calculated as stated in [4].

Sebastian Ohl is with Institute of Control Engineering, Technische Universität Braunschweig, Hans-Sommer-Str. 66, 38106 Braunschweig, Germany ohl@ifr.ing.tu-bs.de

Markus Maurer is with Institute of Control Engineering, Technische Universität Braunschweig, Hans-Sommer-Str. 66, 38106 Braunschweig, Germany maurer@ifr.ing.tu-bs.de

The object based sensor data fusion of the project CarOLO had modeled object hypotheses by open polylines (contours) supplemented by a velocity vector and its derivative. These hypotheses were fused by a contour estimating Kalman filter. The direct usage of this system showed that the increased velocity and road traffic were not tracked robustly in inner-city scenarios. For example, deformations of the contour were observed because of insufficient measurements and wrong velocity vectors. These problems can be overcome by using a simplified modeling of the environment by boxes, as described by [5]. This change of the geometric model would reduce the accuracy of modeling surrounding buildings or non vehicle objects and a great part of the predecessors' flexibility would be lost.

The approach described in this paper does not reduce the flexibility of the contour's description, but it enhances the object hypotheses tracking of most road users by a classification of measurements. By classifying the measurements in manually defined shapes, the minimal needed count of points is achieved. This results in an enhanced tracking of classified object hypotheses, due to the defined forms of the classified shapes, deformation of contours will not occur. Therefore, a contour change has no influence on other state variables.

Compared to the classifier for segmented laser points used by [6], this approach uses the classifier not only to determine reference points for updating the Kalman filter, but also for changing the filter's geometric object hypothesis model. This is similar to the interacting multiple model used by [5]. However, the approach described in this paper changes the model of the object hypothesis and does not calculate all models simultaneously. Only the best fitting model is selected and only this one is calculated by the filter.

This paper is structured as follows: The used sensor configuration is presented in Section II describing the input data. It is followed by Section III illustrating the object hypothesis model used in the Section IV which shows the enhancements of the filters structure. Results of a test run in public traffic are presented in Section V offering an impression of the filter's possibilities.

II. Sensor configuration

The test vehicle in the project Stadtpilot is a 2007 VW Passat called "Leonie". It is equipped with laser and radar sensors (see Figure 1). All sensors are building their own object hypotheses in their ECUs. So, the described filter has to fuse object data only. The vehicle's front is equipped with two IBEO Alaska XT laser scanners, a Hella IDIS fixed multi beam laser sensor and a SMS UMMR 2010 radar sensor. In the vehicles rear, a SMS UMMR Blindspot radar sensor,

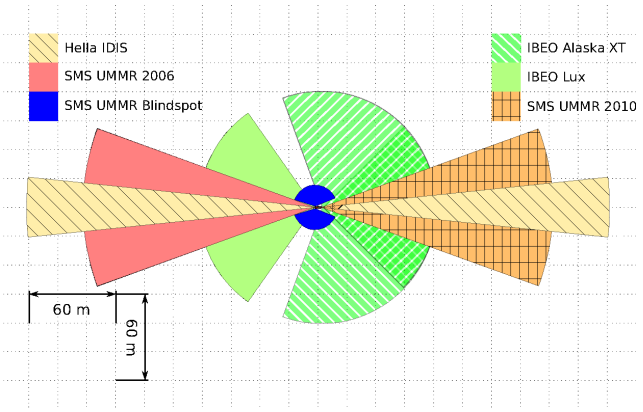

Fig. 1. Sensor configuration of the object based sensor data fusion of the test vehicle Leonie in the project Stadtpilot

another Hella IDIS and a IBEO Lux laser scanner is installed. Other installed sensors, e.g. a Hella IDIS 2 laser scanner are not processed by the object based sensor data fusion and, therefore, not described here.

Depending on the sensor, different preprocessing efforts are conducted by the sensor's control unit. This results in different object hypothesis models. Table I shows the sensor types with the used object hypothesis models.

Sensor	Object hypothesis model
IBEO Alaska XT	open polyline with velocity vector
IBEO Lux	open polyline with velocity vector
Hella IDIS	line
SMS UMMR 2010	point with velocity vector
SMS UMMR 2006	point with radial velocity

TABLE I
OBJECT HYPOTHESIS MODELS OF THE USED SENSORS IN TEST VEHICLE LEONIE

All acquired data can be mapped to an open polyline with or without velocity vector. This enables a uniformed processing of the data by the sensor data fusion and the processing application.

III. OBJECT HYPOTHESIS MODEL

The object hypothesis model of the filter, described in this paper, is related to the coordinated turn model, shown in [7], with a constant angle velocity of $\omega = 0$. It is modeled as an open polyline $(x_0, y_0 \ldots x_n, y_n)$ with a velocity vector consisting of the velocity (v) and the angle of the velocity (α) as well as the according acceleration along the velocity vector (a). Compared to the state vector of CarOLO project's sensor data fusion, the state vector has been reduced by the angle velocity to avoid the special case $\omega = 0$ described in [8] and to reduce the complexity of the matrix operations. Because of this reduction, the object hypothesis' change of orientation is not observed. In practice, no substantial degradation of the other state variables was observed. The state vector is defined as:

$$\hat{x} = [\ x_0 \quad y_0 \quad \ldots \quad x_n \quad y_n \quad \alpha \quad v \quad a\ ]^T \qquad (1)$$

The velocity vector describes the displacement of the complete contour. This conforms to reality only by approximation because parts of the contour can be expanded or reduced. Generally, the correctness of the velocity vector corresponds to the complexity and the change of the contour directly because a higher number of points results in a higher noise level on the contour tracking.

The process model of the state vector $(\hat{x}[k])$ at a discrete point in time (k) is described as:

$$\begin{aligned} x_i[k+1] &= x_i[k] + v[k] * \cos(\alpha[k]) * \Delta t + \\ &\quad \tfrac{1}{2} * a[k] * \cos(\alpha[k]) * \Delta t^2,\ i \in [0..n] \\ y_i[k+1] &= y_i[k] + v[k] * \sin(\alpha[k]) * \Delta t + \\ &\quad \tfrac{1}{2} * a[k] * \sin(\alpha[k]) * \Delta t^2,\ i \in [0..n] \\ \alpha[k+1] &= \alpha[k] \\ v[k+1] &= v[k] + a[k] * \Delta t \\ a[k+1] &= a[k] \end{aligned} \qquad (2)$$

Δt describes the duration of prediction.

IV. FILTER

In the CarOLO project's sensor data fusion, the filter structure contained the stage one (mapping point-to-point) and three (adding points) of the contour point processing supplemented by the extended Kalman filter. Stage two (mapping point-to-line) and the classification of measurements represent the enhancement to the filter described in [3]. The enhanced filter structure is shown in Figure 2.

The process model of the extended Kalman filter is represented by the model discussed in Section III. It describes the motion in the object hypothesis' center of gravity and, therefore, reduces the complexity compared to a model tracking with every point individually. As a result, the complexity is no longer dependent on the number of points.

After mapping the measurement to an object hypothesis already known to the track database, the measurement is classified according to their contour (see Section IV-C). Parallel to this classification, the track is predicted as specified by the process model. In the following, the first two phases of the contour point processing to the points of the measurement are executed. For unclassified measurements, a third phase follows to create new points in the polyline for every not mapped contour point of the measurement. A detailed description of the contour point update algorithm is given in Section IV-A. During the contour point processing, the average deviation $(\Delta\overline{x}, \Delta\overline{y})$ between measurement and track is calculated. These values are passed on to the update step of the Kalman filter.

After updating the state vector by an extended Kalman filter, the contour of the track is updated (see Section IV-B). By calculating the Kalman filter, the offset since the last time step is determined. This offset is summed to the points of the contour. Points of an unclassified track that have not been updated within a certain time are deleted.

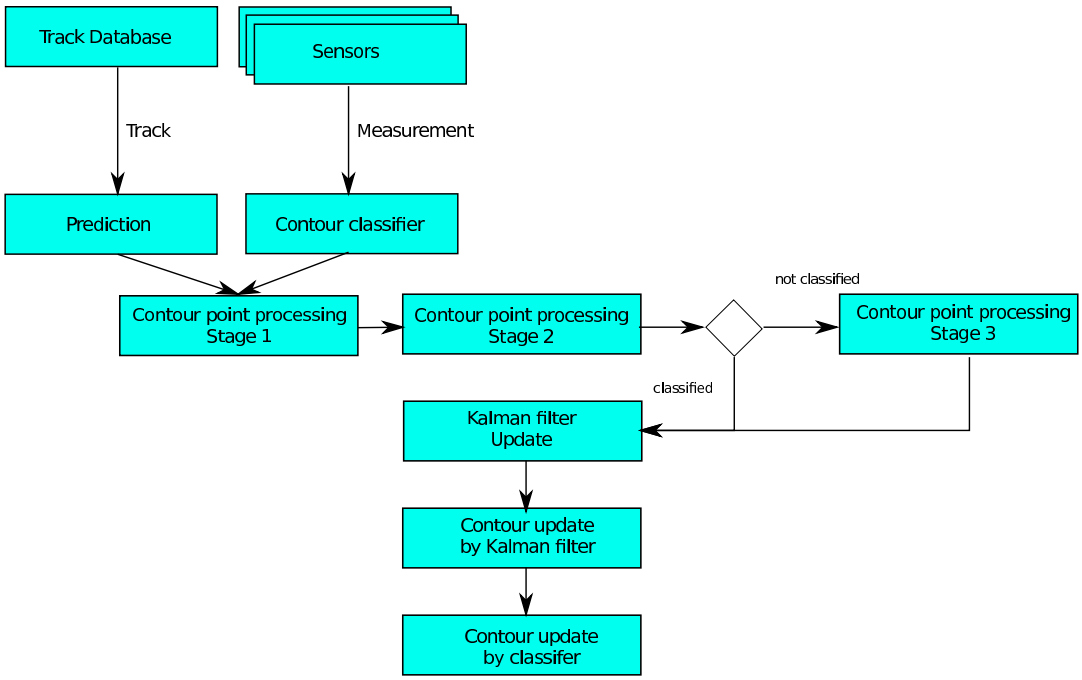

Fig. 2. Structure of the filter

The last step of the filter structure updates the contour by the classification. If the classification has been changed, the contour is substituted by a new one. The classified shape is adapted to size and position of the measurement and inserted into the track instead of the old contour.

A. *Contour point processing*

The extended Kalman filter contains a process model which is reduced to its center of gravity. Therefore, it is not processing each contour point individually. As a result, a mapping from contour points of the measurement to the ones of the track is needed. This process is separated into three stages:

1) Point to point mapping
2) Point to line mapping
3) Adding new points

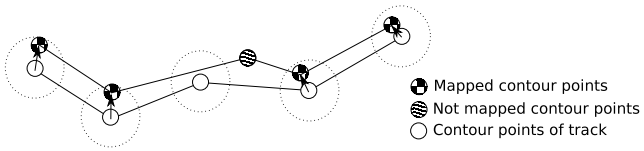

Fig. 3. Contour point processing stage 1

The first stage corresponds to the method described in [3]. Here, one to one mappings between contour points from the measurement to the contour of the track are built (see Figure 3). A gating policy (shown as dotted line), contains the Euclidean distance as metric, as used by e.g. [9], avoid mappings between great distances. After successfully mapping the points, the offset between these points $(\Delta x_i, \Delta y_i)$ is calculated by subtracting the points from each other.

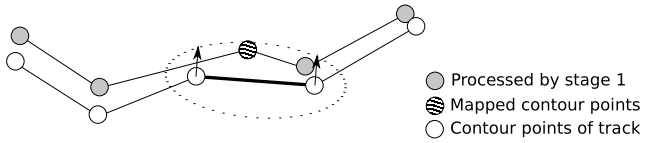

Fig. 4. Contour point processing stage 2

The second stage takes care of points that have not been mapped in stage one. This stage conducts a mapping between a point of the measurement's contour and a line between two points of the track's contour (see Figure 4). This way, a point from another sensor that lies on the line between two points from the track can be used to update the contour without adding a new point to the shape. The perpendicular offset from the measurement point to the line $(\Delta x_i, \Delta y_i)$ is fed to the extended Kalman filter update.

Points that cannot be mapped in the first two stages are added to the track's contour. They do not contribute to the extended Kalman filter update. The third stage is only used for tracks that cannot be classified by the classifier. For already classified object hypotheses the contour point's count is fixed because of the fixed shape. Details are shown in Section IV-C.

B. *Extended Kalman filter*

The state vector of the extended Kalman filter, as depicted in Equation 1, is reduced to the offset of its center of gravity $(\Delta x, \Delta y)$. Thus, the complexity is considerably reduced because the state vector owns a fixed dimension and does not depend on the count of the contour points. The simplified state vector $(\hat{x}'[k])$ is defined as:

$$\hat{x}'[k] = [\ \Delta x \quad \Delta y \quad \alpha \quad v \quad a\ ]^T \tag{3}$$

The process model conforms to the already presented one with $x[k] = 0, y[k] = 0$. The offsets that have been calculated during the processing of the contour points $(\Delta x_i, \Delta y_i)$ of a track and its corresponding measurement are used as measurement vector (z_k) of the extended Kalman filter. Depending on the object hypothesis model of the measurement data (see Table I), the measurement vector is supplemented by a velocity vector. Therefore, z_k is defined as follows:

$$z_k^{Position} = [\ \tfrac{1}{N}\textstyle\sum_{i=1}^{N} \Delta x_i \quad \tfrac{1}{N}\sum_{i=1}^{N} \Delta y_i\ ]^T \tag{4}$$

$$z_k^{Position/Velocity} = [\ \tfrac{1}{N}\textstyle\sum_{i=1}^{N} \Delta x_i \quad \tfrac{1}{N}\sum_{i=1}^{N} \Delta y_i \quad \alpha \quad v\ ]^T \tag{5}$$

Number N correspondents to the number of mappings in stage 1 and 2 of the contour point processing. The Jacobi matrix (H) of the observation matrix (h) correspond to the identity matrix of dimension conforming to the dimension of y. The measurement deviation (R_k) is sensor specific and P_k represents the state's deviation matrix.

After defining the measurement vector and the observation matrix, the update step of the extended Kalman filter can be conducted:

$$\tilde{y}_k = z_k - H_k * \hat{x}'_{k|k-1} \tag{6}$$

$$S_k = H_k * P_{k|k-1} * H_k^T + R_k \tag{7}$$

$$K_k = P_{k|k-1} * H_k^T S_k^{-1} \tag{8}$$

$$\hat{x'}_{k|k} = \hat{x}'_{k|k-1} + K_k * \tilde{y}_k \tag{9}$$

$$P_{k|k} = (I - K_k * H_k) * P_{k|k-1} \tag{10}$$

The Kalman gain (K_k) is calculated only once for all contour points and assumed equal for all points. The update of a single contour point with a mapping to a measurement is, therefore, defined as follows:

$$x_{i,k|k} = x_{i,k|k-1} + K_k * \Delta x_{i,k} \tag{11}$$

$$y_{i,k|k} = y_{i,k|k-1} + K_k * \Delta y_{i,k} \tag{12}$$

Not mapped contour points are updated by the values of $\hat{x}'_{k+1|k+1}\{1, 2\}$.

C. Contour classifier

The core of the filter's enhancement to the fusion system of the CarOLO project is the contour classifier. It estimates the shape of the measurement and replaces the track's contour on a successful classification. This results in a simplified description and, therefore, in a lower computing time and improved tracking.

Common shapes from the test area on the inner-city ring able to influence their behavior on their own have been chosen for this classification. Due to the object hypotheses created by the sensors, the shapes depicted in Table II have been chosen to be classified.

Point	•	
L-shape (to the right)	⌊	
Line	|	
U-shape	⌴	
L-shape (to the left)	⌋	
Unknown	N	

TABLE II
CONTOURS CHOSEN FOR CLASSIFICATION

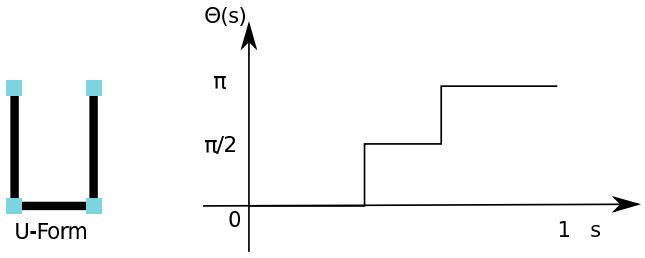

Fig. 5. Turning function $(\Theta_A(s))$ of an U-shape (A)

With the algorithm described in [10], shapes can be compared independently on their scale, rotation, and position by standardization. In this context, the turning function $(\Theta_A(s))$ is used. It measures the angle between the legs of two lines of a polyline (A) in anti-clockwise direction as a function over the length of the polyline (s). By scaling the polyline's length to value 1, the independence of the expanse is achieved. Figure 5 shows the turning function of a U-shape. By calculating the turning function for two polylines A and B and combining them with an integral, a similarity value for these polylines can be calculated:

$$D_B^A = \int_0^1 |\Theta_A(s) - \Theta_B(s)| ds \tag{13}$$

To fuse the similarity values of the measurement with the ones of the track, a Dempster-Shafer-filter (D-S), as described by [11], is used. To do so, a frame of discernment (2^Θ) containing the elements of the shapes to classify and supplemented by a not classified shape (N) is defined:

$$2^\Theta_{Polyline} = \{N, \bullet, |, \lrcorner, \llcorner, ⌴\} \tag{14}$$

The associated mass distribution (M) of the D-S

$$\begin{array}{llll} M(N) & \in & [0,1] & M(\lrcorner) \in [0,1] \\ M(\bullet) & \in & [0,1] & M(\llcorner) \in [0,1] \\ M(|) & \in & [0,1] & M(⌴) \in [0,1] \end{array}$$

$$M(N) + M(\bullet) + M(|) + M(\lrcorner) + M(\llcorner) + M(⌴) = 1$$

results from the frame of discernment and is added to the state vector of every track. By using the similarity values of the classifier (D_B^A) and scaling them to 1, a second mass distribution can be created. The masses of the track and the masses of the classifier can be combined by using the D-S-rule-of-combination. As a result, a new mass distribution (M_C) is calculated:

$$M_C = M(C|A,B) = \frac{1}{1-\kappa} \sum_{i,j|A_i \cap B_j = C} M_A(A_i) * M_B(B_j) \quad (15)$$

$$\kappa = \sum_{i,j|A_i \cap B_j = \emptyset} M_A(A_i) * M_B(B_j) \quad (16)$$

After combining the masses, the one with the greatest evidence can be selected. For the determination of the polyline's similarity values, the polyline have to consist of at least three points. Therefore, the determination of the frame of discernment's masses cannot be done for all measurement data. To process object hypotheses described by a point or line object hypothesis model additional frames of discernment have to be defined:

$$2^{\Theta}_{Point} = \{N, (\bullet, |, \rfloor, \lfloor, \sqcup)\} \quad (17)$$

$$2^{\Theta}_{Line} = \{N, \bullet, (|, \rfloor, \lfloor, \sqcup)\} \quad (18)$$

Here, the masses that cannot be described by the sensor's object hypothesis model are merged to one mass because an object hypothesis described by a point can be derived from a U-shape in reality. The same applies to object hypotheses with a line object hypothesis model. Lines with a very short length are treated as they describe a point and, therefore, can support the point mass. At a certain length, the modeled line may be part of a more complex contour and, therefore, not be completely described by the sensor's object hypothesis model.

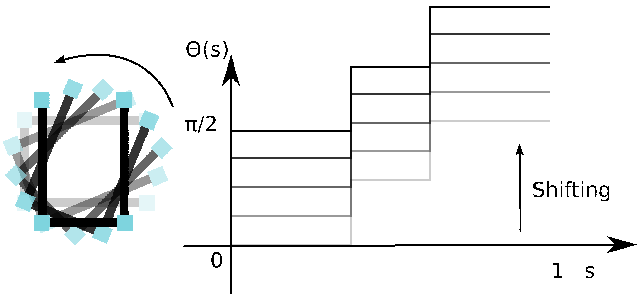

Fig. 6. Finding the correct rotation angle by minimizing the similarity integral ($D_B^{A'}$). Rotating shape A' until it matches shape B (left), shifting the turning function (right)

The contour tracked by the extended Kalman filter is replaced if a change in the contour classifier occurs. Then, the classified shape is scaled, rotated, and translated to the tracked position. To calculate the optimal angle for rotation, the contour classifier is used. By defining a virtual line to the first contour point of shape A (A'), the function $\Theta_{A'}(s)$ is no longer rotation invariant because this addition results in a shift on the y-axis. As depicted in Figure 6, the optimal angle can be calculated by scanning the range $[-\pi, \pi)$ and minimizing the integral $D_B^{A'}$. The Figure shows a U-form which needs to be rotated by $90°$. The turning function is calculated several times for the rotated and the unrotated shape. If Equation 13 is minimized, the optimal angle for rotating the classified shape is found.

In the following, the contour is processed by stage 1 and 2 of the contour point processing only. New points are not added because the basic shape of the contour is defined by the classifier. Possible variations, e.g. deformations of a L-shape, can be adjusted by stage 1 of the contour point processing.

V. Results

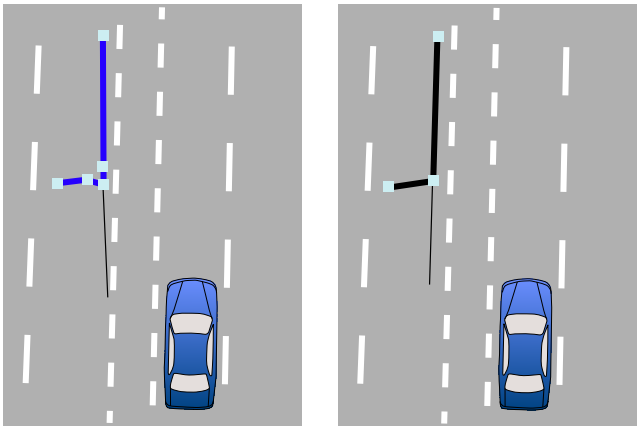

Fig. 7. Comparison of measurement with classified track. Measurement of IBEO Alaska XT (left), Track (right)

Figure 7 shows a situation that has been recorded during a test run on Braunschweig's city ring. The test vehicle Leonie drives on the right lane while on the opposite lane another vehicle is passing by. The contour of the other vehicle is described by five contour points by the sensor. The D-S classifies this measurement as a "L-shape (left)" and, therefore, it is replaced by this shape. The slight opening of the L-shape is caused by the first stage of the contour point processing. The lower count of contour points results in a less noise-prone track. Therefore, the estimation of the velocity vector is improved. As depicted in Figure 7, the direction of the vector is aligned to the lane in the aerial picture whereas the vector of the measurement differs by $\approx 4°$.

Figure 8 shows the count of contour points of an overtaking vehicle with and without classifier over a period of time. While the tracking with enabled classifier is able to track the target from approx. 4.5s stably (mark 1), the track with disabled classifier is lost at approx. 5.6s (mark 2) and a new track has to be initialized. Figure 8 shows only the new one. Comparing the count of the contour points with enabled and disabled classifier, the reduction of the contour points is revealed. One can see the change of the track's contour. Initially, the target is represented as point shape (mark 1), then the classification is changed to a line shape

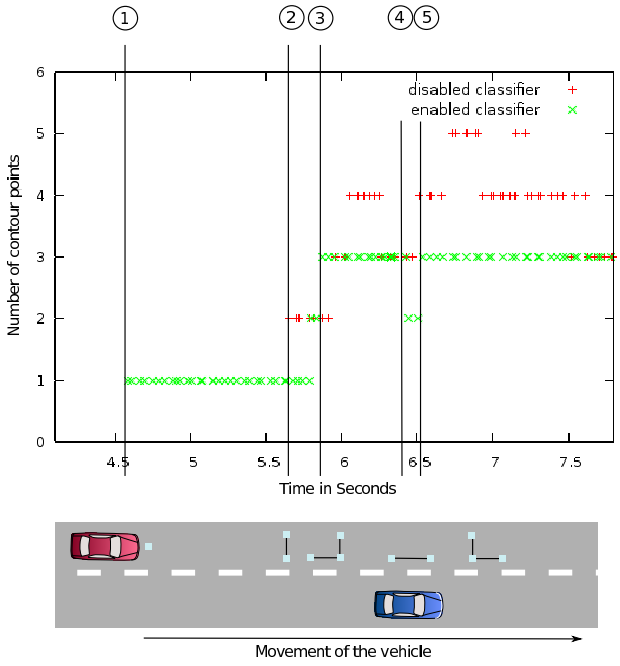

Fig. 8. Count of contour points over a period of time

	MSE
Enabled Classifier	0.0093
Disabled Classifier	0.0158

TABLE III
RMSE OF AN SIMULATED OVERTAKING SITUATION

described by two points (mark 2), and finally to a L-shape described by three points (mark 3). As both vehicles are on the same height (approx. 6.5s), the track is classified as a line shape again (mark 4), and as a L-shape afterwards (mark 5). Without classifier the contour is represented by four or temporarily five contour points.

To get a clear reference data of an overtaking situation, it was simulated with our project specific sensor data simulator. The data is comparable to the time frame, running from mark 3 to the end point in Figure 8. The overtaking maneuver takes place at a relative velocity of 10 m/s and consumes 4 seconds. The simulated sensor has an update rate of 10 Hz. The processed contours contain 3 or 5 points and result in contours with 2 or 3 points with enabled classifier, and 3 or 5 points with disabled classifier. To every contour point, Gaussian white noise with a standard deviation of 0.2m is added. This value of 0.2m was determined experimental for these situations. Referring to Table III, the classifier approach offers a significant benefit over the non-classifier approach as it provides a reduced mean square error (MSE).

VI. Summary and Outlook

This paper has introduced a contour classifying Kalman filter for an urban environment. It classifies measurement data based on the turning function and fuse the result by a Dempster-Shafer-filter. If the classification changes, the contour of the track is replaced by the classified shape after the filter's update. This results in a considerable enhancement of the representation of frequently emerging shapes without reducing the flexibility of the free-form-object hypothesis model for unclassified object hypotheses.

The method described above is currently used and will be further developed in the project Stadtpilot for autonomous test drives in inner-city areas. One possibility for an enhancement is to classify contour parts. By combining them to complex contours currently unclassified objects could be described effectively. Another possibility is to use the sensor's own classifier (e.g. pedestrian, trucks etc.) and fuse this classification with the D-S. Furthermore, the replacement of the contours by its basic classified shape could be improved by a better approximation of the rotation and scale. To determine the quality of this approach, a comparison of an interacting multiple model filter with various geometric object hypothesis models is subject for further studies.

VII. Acknowledgments

The authors gratefully acknowledge the contribution of all team members taking part in the *Stadtpilot* project.

References

[1] T. Nothdurft, P. Hecker, S. Ohl, F. Saust, M. Maurer, A. Reschka, and J. R. Böhmer, "Stadtpilot: First fully autonomous test drives in urban traffic," in *14th International IEEE Annual Conference on Intelligent Transportation Systems*, Washington DC, United States, 2011.

[2] F. W. Rauskolb, K. Berger, C. Lipski, M. Magnor, K. Cornelsen, J. Effertz, T. Form, F. Graefe, S. Ohl, W. Schumacher, J.-M. Wille, P. Hecker, T. Nothdurft, M. Doering, K. Homeier, J. Morgenroth, L. Wolf, C. Basarke, C. Berger, T. Gülke, F. Klose, and B. Rumpe, "Caroline: An autonomously driving vehicle for urban environments," *Journal of Field Robotics*, vol. 25, no. 9, pp. 674–724, August 2008.

[3] J. Effertz, "Autonome Fahrzeugführung in urbaner Umgebung durch Kombination objekt- und kartenbasierter Umfeldmodelle (translated: Autonomous vehicle guidance in urban environments by combining object- and gridbased environment models)," Ph.D. dissertation, Institute of Control Engineering, Technische Universität Braunschweig, Braunschweig, Germany, February 2009.

[4] Y. Bar-Shalom, T. Kirubarajan, and X.-R. Li, *Estimation with Applications to Tracking and Navigation*. New York, NY, USA: John Wiley & Sons, Inc., 2002.

[5] N. Kämpchen, "Feature-level fusion of laser scanner and video data for advanced driver assistance systems," Ph.D. dissertation, Universität Ulm, 2007.

[6] S. Wender, "Multisensorsystem zur erweiterten Fahrzeugerfassung (translated: Multi sensor system for enhanced automotive perception," Ph.D. dissertation, Ulm University, 2007.

[7] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Norwood: Artech House Publishers, 1999.

[8] J. Effertz, "Sensor architecture and data fusion for robotic perception in urban environments at the 2007 DARPA urban challenge," in *RobVis'08: Proceedings of the 2nd international conference on Robot vision*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 275–290.

[9] J.-C. Becker, "Fusion der Daten der objekterkennenden Sensoren eines autonomen Straßenfahrzeugs (translated: Sensor data fusion of the data of the object based sensors of an autonous road vehicle)," Ph.D. dissertation, Institute of Control Engineering, Technische Universität Braunschweig, March 2002.

[10] E. M. Arkin, L. P. Chew, D. P. Huttenlocher, K. Kedem, and J. S. B. Mitchell, "An efficiently computable metric for comparing polygonal shapes," in *SODA '90: Proceedings of the first annual ACM-SIAM symposium on Discrete algorithms*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1990, pp. 129–137.

[11] H. Wu, M. Siegel, R. Stiefelhangen, and J. Yang, "Sensor fusion using dempster-shafer theory," in *IEEE International Measurement Technology Conference*, Anchorage AK, USA, 2002.